導入

杉山将・本多淳也
sugi@k.u-tokyo.ac.jp, jhonda@k.u-tokyo.ac.jp
http://www.ms.k.u-tokyo.ac.jp

講義の流れ

1. 授業概要
2. 確率・統計の基礎
   A) 確率変数と確率分布
   B) 多変数の確率分布
   C) 確率変数の性質を表わす指標
3. Octaveの使い方

先端データ解析論

- **内容**：機械学習の主要技術を紹介する
- **講義資料**：
  - 以下のITC-LMSのサイトで公開する

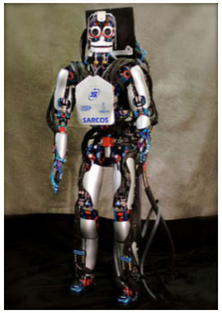
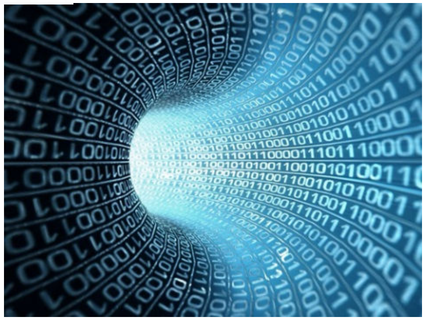
https://itc-lms.ecc.u-tokyo.ac.jp/lms/course/view.php?id=89681

- **成績評価**：
  - ほぼ毎週出す小レポート課題
    （次回の授業までにITC-LMS上でpdf形式で提出）
  - 講義中にたまに行う小さな演習
    （発表した学生には加点する）

機械学習

- **機械学習**(⊂人工知能？)：データの背後に潜む知識を学習する
- **様々な応用例**：
  - 音声・画像・動画の認識
  - ウェブやSNSからの情報抽出
  - 商品やサービスの推薦
  - 工業製品の品質管理
  - ロボットシステムの制御
- **ビッグデータ**時代の到来に伴い，機械学習の重要性は益々高まりつつある




教科書

- 杉山：イラストで学ぶ機械学習，講談社，2013
  - 最小二乗回帰，スパース回帰，ロバスト回帰
  - 最小二乗分類，サポートベクトル分類，確率的分類，系列データの分類
  - オンライン学習，マルチタスク学習，半教師付き学習，転移学習
  - 線形次元削減，カーネルを用いた非線形次元削減，ニューラルネットワークを用いた非線形次元削減

参考書

- Bishop著，元田ほか訳：パターン認識と機械学習（上・下），丸善出版，2007.
- Hastieほか著，杉山ほか訳：統計的学習の基礎，共立出版，2014.
- 杉山，機械学習のための確率と統計，講談社，2015.
- 杉山：統計的機械学習，オーム社，2009.
- 八谷，杉山：強くなるロボティック・ゲームプレイヤーの作り方，毎日コミュニケーションズ, 2008.

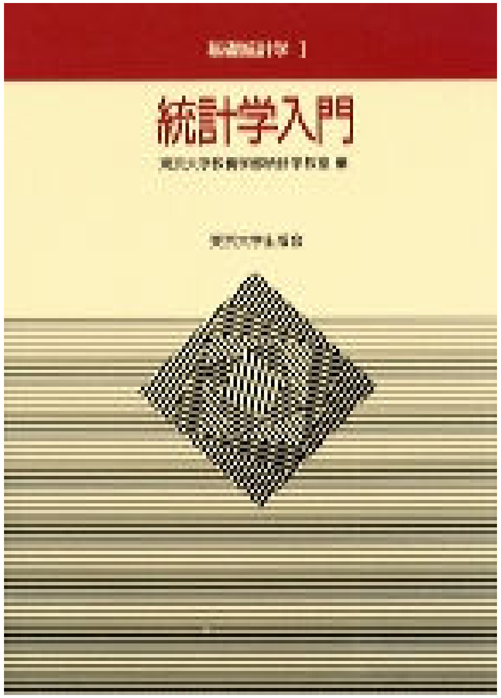
機械学習をより深く学びたい人へ：機械学習プロフェッショナルシリーズ

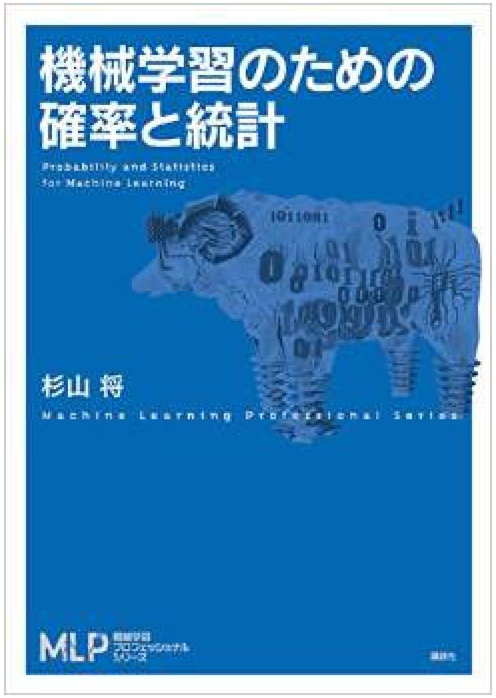
講義の流れ

1. 授業概要
2. 確率・統計の基礎
   A) 確率変数と確率分布
   B) 多変数の確率分布
   C) 確率変数の性質を表わす指標
3. Octaveの使い方

参考書

- 東京大学教養学部統計学教室(編), 統計学入門, 東京大学出版会, 1991年
- 杉山将, 機械学習のための確率と統計, 講談社, 2015年





確率変数と確率分布

- 確率変数(random variable):とる値に対して確率が与えられている変数
- 実現値:確率変数が実際にとる値
- 確率分布(probability distribution):確率変数の実現値と確率との関係を関数として表現したもの

- 確率変数は大文字で，実現値は小文字で表わすことが多い(が，そうでない場合も多い)

- カテゴリy:離散型の確率変数
- パターンx:連続型の確率変数

離散型の確率変数と確率関数

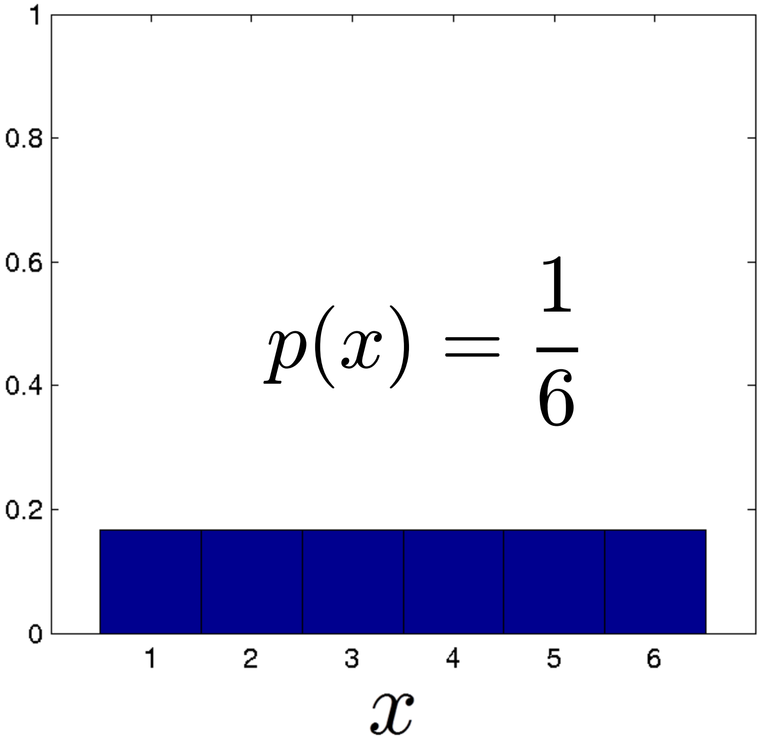
- 離散型(discrete type)確率変数: 可算集合の中の値をとる確率変数
- 離散型の確率変数の確率分布: 確率変数がそれぞれの値をとる確率

$$\Pr(Y = y) = p(y)$$

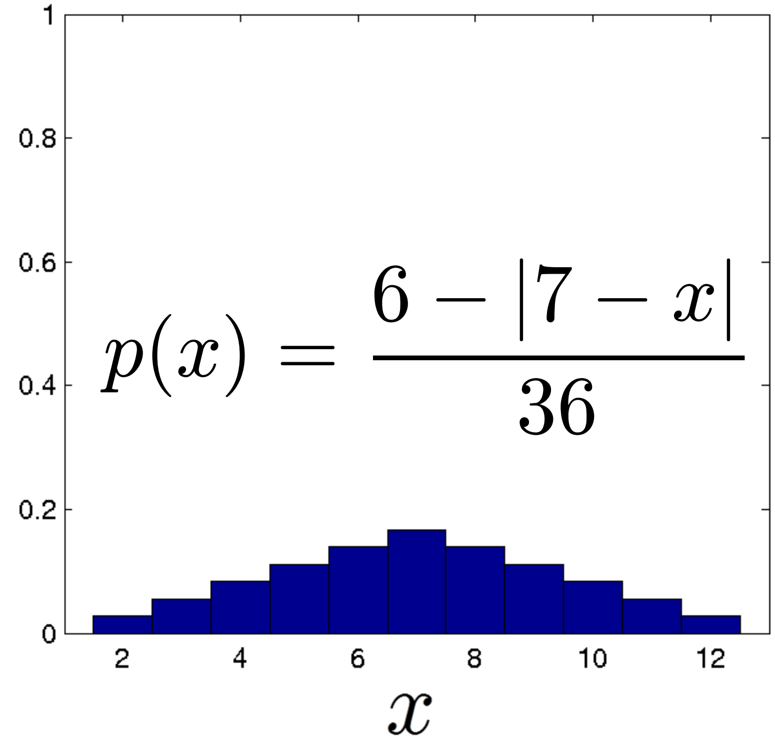
$p(y)$:確率質量関数(probability mass function)

$$p(y) \geq 0, \qquad \sum_y p(y) = 1$$

離散型の確率分布の例



さいころの出る目

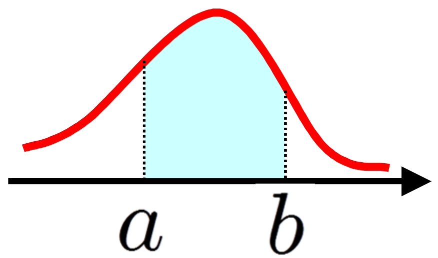


二つのさいころの
出る目の和

連続型の確率変数と確率密度関数

- 連続型(continuous type)確率変数:連続値をとる確率変数
- 連続型の確率変数の確率分布:確率変数が a 以上 b 以下の値をとる確率

$$\Pr(a \leq X \leq b) = \int_a^b p(x)dx$$



$p(x)$:確率密度関数(probability density function)

$$p(x) \geq 0, \int p(x)dx = 1$$

- 注意:連続型の確率変数がある値 a をとる確率はゼロ!

$$\Pr(X = a) = \int_a^a p(x)dx = 0$$

講義の流れ

1. 授業概要
2. 確率・統計の基礎
   A) 確率変数と確率分布
   B) 多変数の確率分布
   C) 確率変数の性質を表わす指標
3. Octaveの使い方

同時確率

- $p(x,y)$: X と Y の同時確率(joint probability)
- 周辺化(marginalization):

$$\sum_y p(x,y) = \underbrace{p(x)}_{\text{周辺確率 (marginal probability)}} \quad \text{または} \quad \int_D p(x,y)dx = p(y)$$

- より正確には同時確率は $p_{X,Y}(x,y)$,
  周辺確率は $p_X(x)$, $p_Y(y)$ と表記するが,
  省略されることが多い
- 独立性: X と X' が独立(independent):

$$p(x,x') = p(x)p(x')$$

条件付き確率

- $p(x \mid y)$: 条件付き確率(conditional probability)
  - y が起こったもとで x が起こる確率

- 同時確率は

$$p(y \mid x)p(x) = p(x, y) = p(x \mid y)p(y)$$

と表せるので

$$p(x|y) = \frac{p(x,y)}{p(y)} \ \text{for} \ p(y) \neq 0$$

$$p(x) = \sum_y p(x,y)$$

ベイズの定理

- ベイズの定理(Bayes' theorem):

$$p(y \mid x) = \frac{p(x \mid y)p(y)}{p(x)}$$

- 事前確率(a priori probability) $p(y)$ ：
  パターンを知る前のカテゴリの出現確率
- 事後確率(a posteriori probability) $p(y \mid x)$ ：
  パターンを知った後のカテゴリの出現確率

ベイズの定理の使用例

- 結構精度の良いウソ発見器:
  - X:被験者の発言がウソか本当か
  - Y:ウソ発見器の出力
  - ウソをウソと判定する確率:

    p(Y=ウソ | X=ウソ)=0.99
  - 本当を本当と判定する確率:

    p(Y=本当 | X=本当)=0.95
- 被験者がウソをつく確率が

  p(X=ウソ)=0.001

  のとき,ウソ発見器がウソだと言ったら信用すべきか?

ベイズの定理の使用例

- ベイズの定理を用いると

p(X=ウソ|Y=ウソ)=p(Y=ウソ|X=ウソ)p(X=ウソ)/p(Y=ウソ)

- p(Y=ウソ|X=ウソ)=0.99
- p(X=ウソ)=0.001
- P(Y=ウソ)= p(X=ウソ) p(Y=ウソ|X=ウソ)
  + p(X=本当) p(Y=ウソ|X=本当)
  =0.001 x 0.99
  + 0.999 x 0.05
  ≒0.051

→ p(X=ウソ|Y=ウソ)≒0.99 x 0.001/0.051≒0.019
→ p(X=本当|Y=ウソ)=1-p(X=ウソ|Y=ウソ)≒0.981

ベイズの定理の使用例

$p(X=\text{ウソ} \mid Y=\text{ウソ}) \fallingdotseq 0.019$
$p(X=\text{本当} \mid Y=\text{ウソ}) \fallingdotseq 0.981$

- $p(X=\text{ウソ} \mid Y=\text{ウソ}) \ll p(X=\text{本当} \mid Y=\text{ウソ})$なので,
ウソ発見器の結果は信用すべきでない！

- ウソつきが少ない場合はウソ発見器は信用できない
- 実際, $p(X=\text{ウソ} \mid Y=\text{ウソ}) > p(X=\text{本当} \mid Y=\text{ウソ})$が成り立つためには,

$$p(X=\text{ウソ})>0.048$$

である必要がある

講義の流れ

1. 機械学習でどんなことができるか？
2. 確率・統計の基礎
   A) 確率変数と確率分布
   B) 多変数の確率分布
   C) 確率変数の性質を表わす指標
3. Octaveの使い方

確率変数の性質を表わす指標

- 期待値(expectation):確率変数の値の平均（正確には確率による重み付きの平均）
- 確率変数 X の期待値を $E(X)$ で表す

  - 離散型: $$E(Y) = \sum_y y p(y)$$
  - 連続型: $$E(X) = \int x p(x) dx$$

期待値演算の性質

- 定数は期待値をとっても値は変わらない:

$$E(c) = c$$

- 和の期待値は，期待値の和と等しい:

$$E(X + X') = E(X) + E(X')$$

- 定数倍の期待値は，期待値の定数倍と等しい:

$$E(cX) = cE(X)$$

証明は宿題

⇒ 期待値演算は線形

確率変数のばらつきの指標

- 期待値は確率変数の代表する値を表す指標
- 分散(variance): 確率変数の散らばり具合を表す指標

$$V(X) = E\{(X - E[X])^2\}$$

  - 標準偏差は分散の平方根 $\sqrt{V(X)}$

- 分散は次式の方が計算しやすいこともある

$$V(X) = E\{X^2 - 2XE[X] + (E[X])^2\}$$

$$= E[X^2] - 2E[X]E[X] + (E[X])^2\}$$

$$= E[X^2] - (E[X])^2$$

分散演算の性質

1. 定数の分散はゼロ

$$V(c) = 0$$

2. 定数を足したものの分散は，もとの分散と等しい

$$V(X + c) = V(X)$$

3. 定数倍の分散は，もとの分散に定数の2乗をかけたものと等しい

$$V(cX) = c^2 V(X)$$

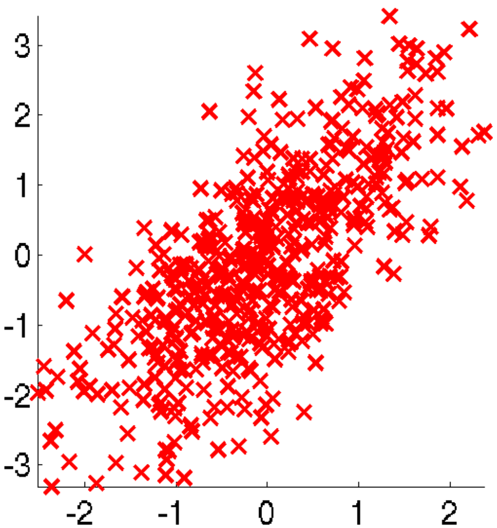
証明は宿題

共分散

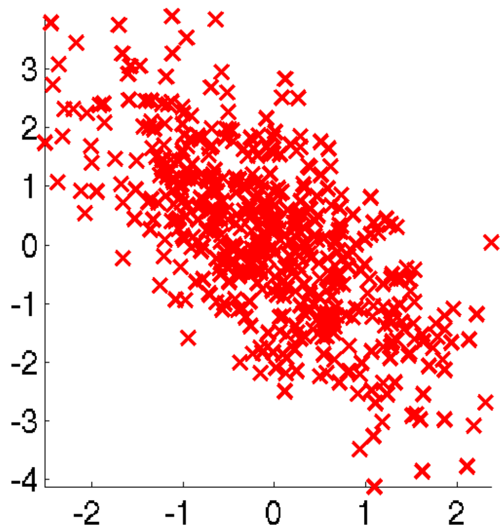
- XとX'の共分散(covariance):

$$\mathrm{Cov}(X, X') = E[(X - E[X])(X' - E[X'])]$$

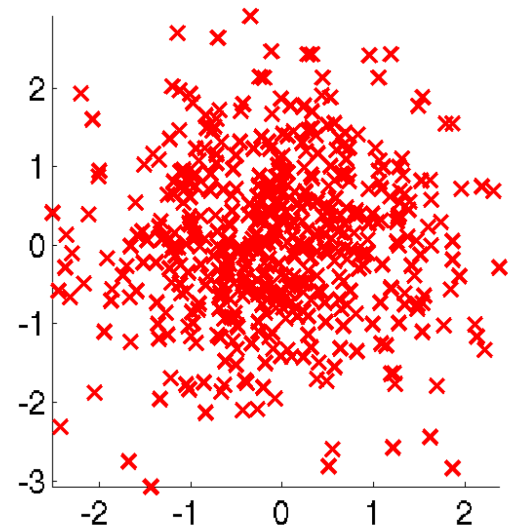
  - $\mathrm{Cov}(X, X') > 0$ のとき，X とX'の増減は同傾向
  - $\mathrm{Cov}(X, X') < 0$ のとき，X とX'の増減は逆傾向
  - $\mathrm{Cov}(X, X') = 0$ のとき，X とX'の増減は無関係



$\mathrm{Cov}(X, X') > 0$



$\mathrm{Cov}(X, X') < 0$



$\mathrm{Cov}(X, X') \approx 0$

分散と共分散の性質

1. 二つの確率変数 X と X' の和の期待値は，それぞれの期待値の和と等しい：

$$E(X+X') = E(X) + E(X')$$

2. しかし X と X' の和の分散は，一般にはそれぞれの分散の和とは等しくない．

$$V(X+X') = V(X) + V(X') + 2\mathrm{Cov}(X, X')$$

$$\mathrm{Cov}(X, X') = E[(X - E[X])(X' - E[X'])]$$

証明は宿題

共分散の使用法の例

$$V(X+X') = V(X) + V(X') + 2\mathrm{Cov}(X, X')$$

- A社の株価を X, B社の株価を X' とする
- $\mathrm{Cov}(X, X') > 0$ のとき,

$$V(X+X') > V(X) + V(X')$$

  - A, B両社の株を買うと, 分散が拡大
  - 変動リスクが増大し, 資産価値は不安定

- $\mathrm{Cov}(X, X') < 0$ のとき,

$$V(X+X') < V(X) + V(X')$$

  - A, B両社の株を買うと, 分散が縮小
  - 変動リスクが抑制され、資産価値は安定

分散共分散行列

- 分散共分散行列(variance-covariance matrix):

$$\mathbf{\Sigma} = E\left[\left\{\begin{pmatrix} X \\ X' \end{pmatrix} - E\begin{pmatrix} X \\ X' \end{pmatrix}\right\}\left\{\begin{pmatrix} X \\ X' \end{pmatrix} - E\begin{pmatrix} X \\ X' \end{pmatrix}\right\}^{\top}\right]$$

$$= \begin{pmatrix} V(X) & \mathrm{Cov}(X, X') \\ \mathrm{Cov}(X, X') & V(X') \end{pmatrix}$$

- 対角成分は分散, 非対角成分は共分散.
- 単に共分散行列ともよぶ

相関

- 相関係数(correlation coefficient): 共分散を標準偏差で割った値

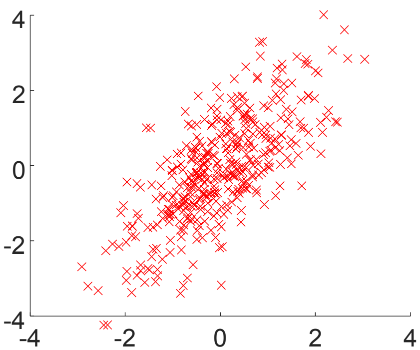
$$\rho_{XY} = \frac{\mathrm{Cov}(X,Y)}{\sqrt{V(X)}\sqrt{V(Y)}}$$

- 相関係数は $-1 \le \rho_{XY} \le 1$ を満たす.

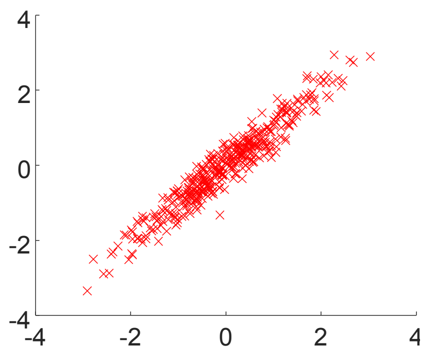
- $\rho_{XY} > 0$ のとき，正の相関があるという.
- $\rho_{XY} < 0$ のとき，負の相関があるという.
- $\rho_{XY} = 0$ のとき，無相関(uncorrelated)であるという.
- $|\rho_{XY}| \approx 1$ のとき，X と Y の増減関係はより確定的になる.

相関係数

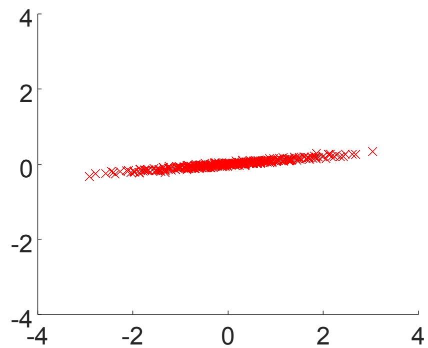
- 相関係数はスケール変換に対して不変，
また非線形の影響を必ずしも反映しない



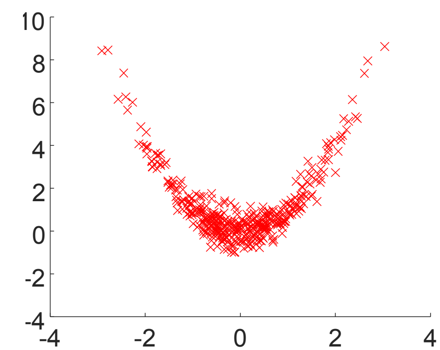
$\rho_{XY} \approx 0.7$



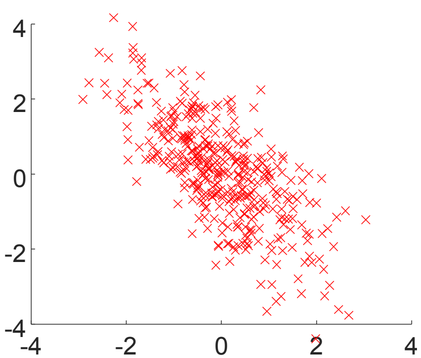
$\rho_{XY} \approx 0.95$



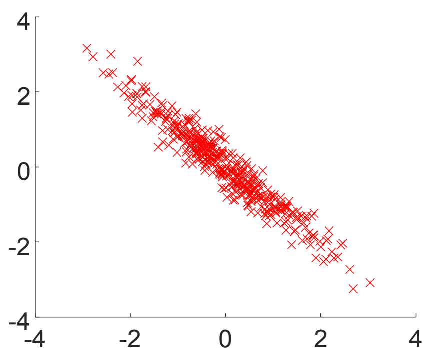
$\rho_{XY} \approx 0.95$



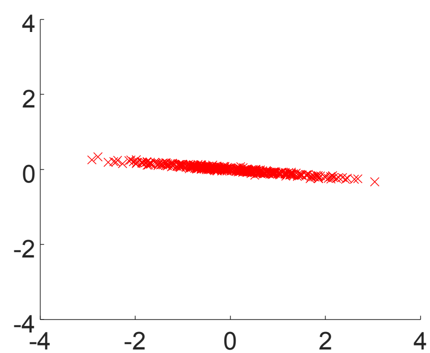
$\rho_{XY} \approx 0$



$\rho_{XY} \approx -0.7$



$\rho_{XY} \approx -0.95$



$\rho_{XY} \approx -0.95$

講義の流れ

1. 授業概要
2. 確率・統計の基礎
3. Octaveの使い方

Octaveとは

- Octave：数値計算用インタプリタ言語
  - “MATLABのフリー版”
  - 行列やベクトルの演算，連立方程式，固有値，統計計算などが簡単に行える
  - グラフ表示が簡単に行える（gnuplotを利用）
- 以下から無料でダウンロードできる：

https://www.gnu.org/software/octave/download.html

Octaveの起動

- Octaveのアイコンをダブルクリックすれば, Octaveが起動する.
- まずはOctaveを電卓のように使ってみよう.

```
> (1+2)*(3-4)*5^2
ans = -75

>exp(10)*log(5)*sin(0.5*pi)
ans =  3.5450e+004

> exit
```

- exitはOctaveを終了するコマンドである.

ベクトル，行列の演算

- Octaveの特徴は，ベクトルや行列を直接扱うことができるところである.
- ベクトルや行列は四角括弧で定義する.
- 横方向の要素はスペースで区切り，縦方向の要素はセミコロンで区切る.

```
> A=[1 2;3 4]
A =
   1   2
   3   4
```

```
> b=[3;4]
b =
   3
   4
```

ベクトル，行列の演算

- ベクトルや行列の演算はそのままスカラーの場合と同じように行う.
- 行末にセミコロンをつけると値を表示しない.
- 単に変数名を入力すれば変数の値が表示される.

```
> A*b
ans =
   11
   25
```

```
> C=A*b;
```

```
> C
C =
   11
   25
```

ベクトル，行列の演算

- ベクトルや行列の転置はアポストロフィをつける.
- 行列の要素を取り出すときは，丸括弧で要素を指定する.
- 行列の要素に値を直接代入することもできる.

```
> C'
ans =
  11  25
```

```
> A(2,1)
ans = 3
```

```
> A(2,1)=5
A =
  1  2
  5  4
```

ベクトル，行列の演算

- 一列（一行）の要素を全て取り出したいときはコロンを用いる.
- ベクトルや行列のスカラ倍は，そのまま記述する.
- 行列の要素同士の掛け算や割り算は「.*」や「./」のようにピリオドをつける.

```
> A(:,1)
ans =
  1
  5
```

```
> D=3*A
D =
    3   6
   15  12
```

```
> A.*D
ans =
    3  12
   75  48
```

```
> A./D
ans =
  0.3333  0.3333
  0.3333  0.3333
```

ベクトル，行列の演算

- 数列のベクトルは以下のように生成する.

```
> e=10:5:30
e =
   10   15   20   25   30
```

- 2番目の引数を省略すれば，公差は1になる.

```
> f=1:4
f =
    1    2    3    4
```

数値演算関数

- Octaveには沢山の数値演算の関数が組み込まれている．例えば，

```
> cos(2/3*pi)
ans = -0.5000
```

- 引数にベクトルや行列を指定すれば，要素ごとにその関数の値を計算する．

```
> sin(f)
ans =
   0.8415    0.9093    0.1411   -0.7568
```

- 他には，cos, tan, exp, log, sqrtなどがある．

数値演算関数

- 各関数の説明はhelpで参照できる．単にhelpと入力すればトピックの一覧が得られる．

```
> help exp
```

- ベクトル要素の最大値，最小値，和，積は，max, min, sum, prodにより得ることができる．

```
> max([1 3 5 2 4])
ans = 5
```

数値演算関数

- 逆行列はinvで求まる(が，多用すべきではない)
- 通常はxA=cの解はc/A, Ay=dの解はA¥dで求める

```
> A=[1 2;3 4];
> c=[1 2]; d=[3;4]

> c*inv(A)
ans =
   1   0
> c/A
ans =
   1   0
```

```
> inv(A)*d
ans =
  -2.0000
   2.5000
> A¥d
ans =
  -2.0000
   2.5000
```

数値演算関数

- 固有値はeigで求める.

```
> [eigvec eigval]=eig(A)
eigvec =
   -0.8246   -0.4160
    0.5658   -0.9094
eigval =
   -0.3723         0
         0    5.3723
```

数値演算関数

- ベクトルの平均，標準偏差，分散はmean, std, varでそれぞれ計算する.

```
> var([1 2 3 4])
ans = 1.6667
```

- ソートはsort関数で行なう.

```
>[sorted index]=sort([7 3 6 1 2])
sorted =
   1   2   3   6   7
index =
   4   5   2   3   1
```

数値演算関数

- 0から1の間の一様乱数はrandで，平均0分散1の標準正規分布に従う乱数はrandnで生成する．引数を指定すれば乱数行列を生成できる．

```
> rand()
ans =  0.63714
> rand(2,2)
ans =
    0.8147    0.1270
    0.9058    0.9134

> randn(1,4)
ans =
    0.3188   -1.3077   -0.4336    0.3426
```

グラフ描画関数

- 標準的な折れ線グラフはplot関数で描画する.

```
> x=[-2:0.1:2]; y=sin(x); plot(x,y)
```

- ヒストグラムはhist関数で描画する.

```
> a=randn(1000,1); hist(a)
```

- 3次元のグラフはsurf, meshで，等高線はcontourで描画する.

```
> x=[-3:0.1:3]; y=x.^2; surf(x'*y);
```

- グラフの保存はprintコマンドで行う.

```
> print -dtiff mygraph.tif
```

Octaveによるプログラミング

- これまでは，コマンドラインに直接コマンドを打ち込むことにより計算を行ってきた.
  より複雑な処理を行うときには，スクリプトや関数を用いる.

スクリプト

- スクリプトは，これまでコマンドラインに打ち込んでいたものをあらかじめテキストファイルに記述したものである.
- 例えば，テキストエディタで次のようなスクリプトを作成し，myscript.mというファイル名で保存する.

```
A=[1 2; 3 4]
b=[5;6]
C=A*b
```

スクリプト

- スクリプトのファイルは現在Octaveが実行されているディレクトリに作成すること.
- 現在のディレクトリは,

```
> pwd
```

により確認できる.

- スクリプトmyscript.mを現在のディレクトリに置いたら, Octaveのコマンドライン上で

```
> myscript
```

と入力することにより, myscript.mに記述されたコマンドを実行できる.

繰り返し

- C言語などと同じように，繰り返しにはfor文を使う.
- 例えば，1から10000までの整数の和を求めるには

```
n=10000;  a=0;
for i=1:n
  a=a+i;
end
```

とすればよい．しかし，for文は計算が遅い.

- 代わりに

```
a=sum(1:1000000);
```

と計算したほうがよい.

条件分岐

- 条件分岐にはif文を使う(左下プログラム参照).
- これも右下のようにすれば高速化できる.

```
x=-10:0.1:10;
for i=1:length(x)
  if x(i)>0
    y(i)=x(i);
  else
    y(i)=-x(i);
  end
end
plot(x,y)
```

```
x=-10:0.1:10;
y=zeros(size(x));
y(x>0)=x(x>0);
y(x<=0)=-x(x<=0);
plot(x,y)
```

- そのほかwhile文やswitch文もある.

関数

- スクリプトの先頭行でfunctionと宣言することにより関数を定義できる.
- 例えば，次のファイルをmyfunc.mという名前で保存する.

```
function y=myfunc(x)

y=x.^2+10*sin(x);
```

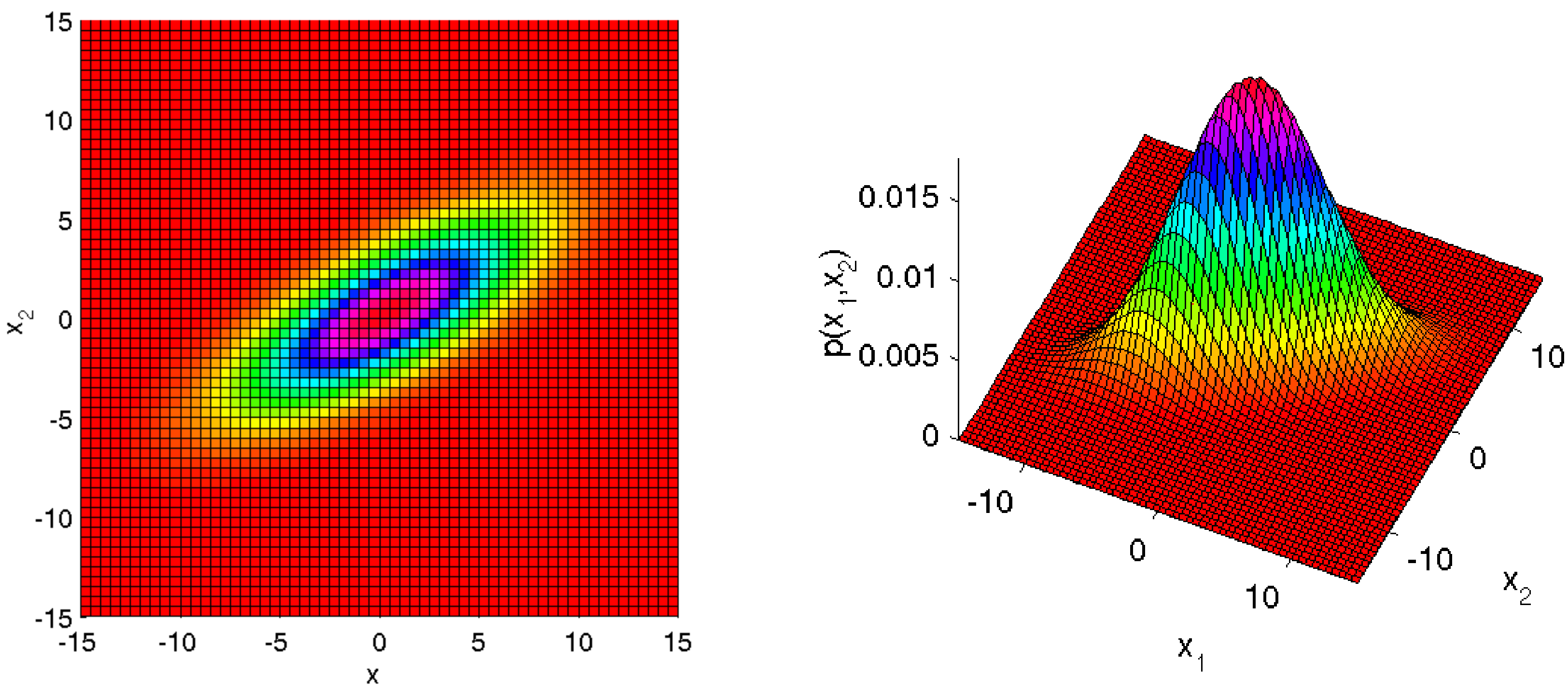
- そして，以下のようにして関数を呼び出す.

```
clear all;
x=[-3:0.1:3]; y=myfunc(x);
figure(1); clf; plot(x,y);
```

例

- 2次元正規分布の確率密度関数の3次元プロットおよび等高線プロットを作成せよ.

$$p(x;\mu,\Sigma)=\frac{1}{(2\pi)^{d/2}\det(\Sigma)^{1/2}}\exp\left(-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right)$$



サンプルプログラム（標準版）


```
clear all
Mu=[0;0]; Sigma=[2 1;1 2];
x1=[-3:0.1:3]; x2=[-3:0.1:3];
for c1=1:length(x1)
  for c2=1:length(x2)
    z=2*pi*sqrt(det(Sigma));
    v=[x1(c1);x2(c2)]-Mu;
    y(c2,c1)=exp(-1/2*v'*inv(Sigma)*v)/z;
  end
end

figure(1); clf; surf(x1,x2,y);
view(45,60)
figure(2); clf; contour(x1,x2,y);
```

サンプルプログラム（高速版）

- 以下のようにするとfor文がなくなり，高速に計算できる

```
clear all
Mu=[0;0]; Sigma=[2 1;1 2];
x1=[-3:0.1:3]; x2=[-3:0.1:3];
X1=repmat(x1-Mu(1),[length(x2) 1]);
X2=repmat(x2'-Mu(2),[1 length(x1)]);
X=[X1(:) X2(:)]';
z=2*pi*sqrt(det(Sigma));
Y=exp(-sum(X.*(inv(Sigma)*X))/2)/z;
y=reshape(Y,[length(x2) length(x1)]);

figure(1); clf; surf(x1,x2,y);
view(45,60)
figure(2); clf; contour(x1,x2,y);
```

宿題1

- 統計の授業が好きな人，嫌いな人の確率が

  $p(X=好)=0.8$　　$p(X=嫌)=0.2$

  であるとする．また，統計の授業が好きな人の中で授業中眠たい人，および，統計の授業が嫌いな人の中で授業中眠たい人の確率が，それぞれ

  $p(Y=眠 \mid X=好)=0.25$　　$p(Y=眠 \mid X=嫌)=0.25$

  であるとする．

  A) $p(X=好, Y=眠)$を求めよ．

  B) $p(Y=眠)$を求めよ．

  C) $p(X=好 \mid Y=眠)$ を求めよ．

  D) 統計の好き嫌いと授業中眠たい事は独立か？

宿題2

- 以下を証明せよ:

A) 定数は期待値をとっても値は変わらない:

$$E(c) = c$$

B) 和の期待値は, 期待値の和と等しい:

$$E(X + X') = E(X) + E(X')$$

C) 定数倍の期待値は, 期待値の定数倍と等しい:

$$E(cX) = cE(X)$$

宿題3

- 以下を証明せよ:

A) 定数の分散はゼロ

$$V(c) = 0$$

B) 定数を足したものの分散は，もとの分散と等しい

$$V(X + c) = V(X)$$

C) 確率変数を定数倍にすると分散の定数の2乗倍に

$$V(cX) = c^2 V(X)$$

D) 和の分散は共分散を用いて以下の式で与えられる

$$V(X + X') = V(X) + V(X') + 2\mathrm{Cov}(X, X')$$

$$\mathrm{Cov}(X, X') = E[(X - E[X])(X' - E[X'])]$$

宿題の提出

- 次回の授業までにITC-LMS上でpdf形式で提出
- 遅れた場合も受け付けるが遅刻として扱う

講義スケジュール

- ４月１１日
- ４月１８日
- ４月２５日
- ５月　２日
- ５月　９日
- ５月１６日
- ５月２３日
- （５月３０日：金曜授業）
- ６月　６日
- ６月１３日
- ６月２０日
- ６月２７日：講義なし
- ７月　４日
- ７月１１日